УДК 514.8

КОНСТРУКТИВНІ ФРАКТАЛИ ЯК РЕЗУЛЬТУЮЧЕ СТИСКУЮЧЕ ВІДОБРАЖЕННЯ ПОДІБНОСТІ.

Адашевська Ірина Юріївна
к.т.н., доцент
Краєвська Олена Олександрівна
доцент
Национальный технічний університет «ХПІ»
м. Харків, Украина

Аннотация: В роботі розглянуто принципи побудови конструктивних фракталів в результаті лінійних (афінних) стискуючих відображень (перетворень) подібності, надано конструкції генераторів (аксіом) фрактальних форм, розглянуто підхід до отримання фрактальних зображень за допомогою системи ітерованих функцій.

Ключевые слова: Фрактал, самоподібність, дробова метрична властивість, L–системи, крива Коха.

Багато об'єктів в природі мають фрактальні властивості, наприклад, узбережжя, хмари, крони дерев, кровоносна система і система альвеол людини або тварин. Фрактали являють собою математичні моделі складних структур, просторове зображення яких представляється у вигляді зламаних, зморшкуватих і нечітких форм [1]. Фрактали (як математичні абстракції) володіють такими характерними властивостями, що відображають їхню іррегулярну сутність:

• самоподоба (ієрархічний принцип організації);
• здатність до розвитку (принцип безперервності формоутворення);
• дробова метрична розмірність (принцип сингулярності міри);
• розмитість, нечіткість контурів (принцип невизначеності кордонів);
• геометричне представлення хаотичної динаміки (принцип динамічного хаосу).

Фрактали, особливо на площині, популярні завдяки поєднанню краси з простотою побудови за допомогою комп'ютера.

Перші приклади самоподібних множин з незвичайними властивостями з'явилися в XIX столітті (наприклад, множина Кантора).

Фрактали зазвичай ділять на

– конструктивні – множини з’являються в результаті лінійних (афінних) стискуючих відображень (перетворень) подібності. Результуюче стискуюче відображення володіє стійкою нерухомою «точкою» - фракталом.

– динамічні - множини виникають в нелінійних (в першу чергу дискретних) динамічних системах. Такі множини можуть володіти масштабною інваріантністю лише наближено.

Конструктивні (класичні) фрактали

Граничного узагальнення ми досягнемо, вибираючи якусь замкнуту криву, яка допускає розбиття на конгруентні сегменти, змінюючи кожен сегмент у який завгодно спосіб так, щоб весь процес можна було повторити на менших сегментах наступних поколінь і перейти до межі. Аналогічні побудови можна виробляти і на поверхнях просторових тіл. Зрозуміло, в результаті таких операцій можуть з’явитися складні криві або поверхні з самоперетинами.

Гарднер

Дуже складні криві можна отримати рекурсивно (ситуація, коли перетворювач прямо або побічно створює власну копію, в теорії алгоритмів називається рекурсією) за допомогою багаторазового «ускладнення» простої кривої. Для опису та побудови подібних кривих використовуються *L*– системи.

L – системи

Поняття *L*– системи тісно пов'язане з самоподібними фракталами.

L– системи використовувалися під час вивчення формальних мов і в біологічних моделях селекції. Вони широке застосовуються в комп'ютерній графіці для побудови фрактальних дерев, рослин і різних фракталів [2].

Для графічної реалізації *L*– систем в якості підсистеми введення використовується так звана черепашача графіка. Точка (черепашка) рухається по екрану дискретними кроками, креслячи свій слід (але може переміщатися і без малювання). Стан черепашки описується трьома параметрами (x, y, α), де (x, y) - координати черепашки, α- напрямок, в якому вона дивиться (рухається). Черепашка інтерпретує і виконує послідовність команд, що задаються кодовим словом, літери якого читаються зліва направо. Кодове слово являє собою результат роботи L– системи і містить такі літери:

F - переміститися вперед на один крок, промальовуючи слід.

b - переміститися вперед на один крок, що не промальовуючи слід (дозволяє організувати розрив в графіку).

+ - збільшити кут α на величину θ (повернути на θ за годинниковою стрілкою)

- - зменшити кут α на величину θ (повернути на θ проти годинникової стрілки)

[- відкрити гілку

] - закрити гілку

X,Y – під час генерування інструкції для пера ігноруються.

Розмір кроку і величина приросту по куту θ задаються заздалегідь і залишаються незмінними для всіх переміщень черепашки. Якщо початковий напрямок руху α (кут, відлічуваний від додатного напрямку осі X) не вказано, але вважають $\alpha = 0$.

Тоді рядок $''F - F + +F - F''$ зі значенням $\theta = 60^\circ$ задає перу команду намалювати першу генерацію K_1 кривої Коха.

Процес перетворення простого рядка в більш складний ґрунтується на наборі інструкцій створення рядків, який входить до підпрограм $producestring(\,)$. Правило для кривої Коха має вигляд:

$'F' \to ''F - F + +F - F''$,

де знак $\to$ означає, що кожен символ F замінюється групою символів $''F - F + +F - F''$. Для символів $'+'$ або $'-'$ інструкція відсутня, тому ці символи передаються без змін.

Формально, детермінована L– система складається з алфавіту, слова ініціації, званого аксіомою (ініаціатором, атомом), і набору породних правил, що вказують, як слід перетворювати слово під час переходу від рівня до рівня (від ітерації до ітерації) (табл.1).

Наприклад, для сніжинки Коха породне правило

$newf = F - F + +F - F;$

Аксіома: $F + +F + +F$ - рівносторонній трикутник; $\theta = \frac{\pi}{3}$.

На першій стадії вихідний рядок називається атомом, в даному випадку це $''F''$, «генерує рядок першого покоління

$S1 = ''F - F + +F - F'',$

потім цей рядок подається на вхід того ж самого процесу, який генерує рядок другого покоління

$S2 = \underline{F - F + +F - F} - \underline{F - F + +F - F} + +\underline{F - F + +F - F} \underline{-F - F + +F - F}$

У цьому рядку можна виділити чотири групи (кластера)

$$F - F + +F - F$$

розділених символами $'-'$, потім $'++'$ і нарешті знову $'-'$. Тепер перо намалює другу генерацію K_2 кривої Коха.

Для сніжинки Коха : $\theta = \frac{\pi}{3}$;Аксіома: $F + +F + +F$ - рівносторонній трикутник; правило: $newf = F - F + +F - F.$

Наведемо алгоритм роботи L– системи в псевдокодах.

Алгоритм L – системи.

Призначення: реалізує правило $F = newf; b = newb$/

Вхід:

$axiom$ - (слово ініціації $F + +F + +F$)

$newf$ - (породне правило) $newf = F - F + +F - F$

$newb$ - (породне правило)

$level$ - (кількість ітерацій, задається).

Вихід:

$word$ - (слово результат)

Ініціація:

$W = axiom$

$n = length\ (W)$ (кількість букв у слові W)

$T = \{\cdot\}$ - (пуста множина)

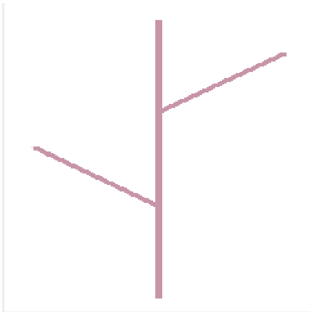


Рис. 1 Крок 1

Кроки:

$white \quad level > 0$

$for\ j = 1\ to\ n$

$if\ W(j) = +, T = \{T +\}, end\ if$

$if\ W(j) = -, T = \{T -\}, end\ if$

$if\ W(j) = [, T = \{T[\}, end\ if$

$if\ W(j) =], T = \{T]\}, end\ if$

$if\ W(j) = F, T = \{Tnewf\}, end\ if$

$if\ W(j) = b, T = \{Tnewb\}, end\ if$

$end\ f\square r\ W = T$

$level = level - 1$

$end\ white$

$word = W.$

Зауваження: $W(j) - j$ –а буква в слові,

$\{T +\}$- рядок T , до якого приєднано знак +.

Галуження

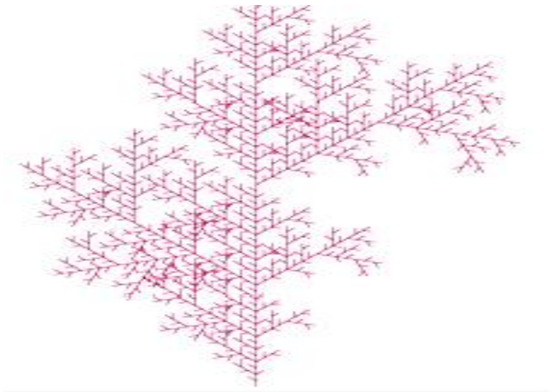


Рис. 2 Крок 5

Для того, щоб запам'ятати бажаний стан, потрібно зробити його копію і записати її в зручному місці. Щоб здійснити це, можна працювати зі стеком. Якщо в рядку зустрічається символ "[" , то поточний стан черепахи заштовхується в стек для подальшого використання. Якщо зустрічається символ"]" , то зі стека виштовхується верхнє значення, і стан черепахи встановлюється таким, що дорівнює виштовхнутому значенню.

{ в *OpenGL* до оператора *switch* підпрограми *produceString ( )* треба додати два рядки:

"[": $save\ Turtle(\,): break$: - заштовхнути в стек поточний стан

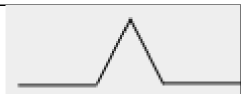
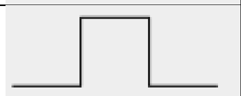
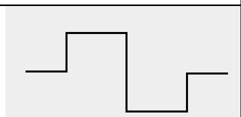
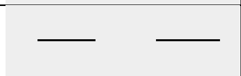
"]": $restoreTurtle\ (\,) : break$:– виштовхнути зі стека стан}

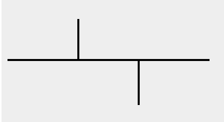
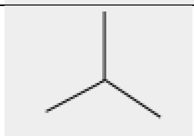
«Кущ». Атом *“F”*,$F \to F[\text{-}F]F[+F][F]; \theta = 51^0$.

На (рис.1), (рис.2) представлені перший і п'ятий крок.

Таблиця 1

Конструкції генераторів (аксіом) фрактальних форм

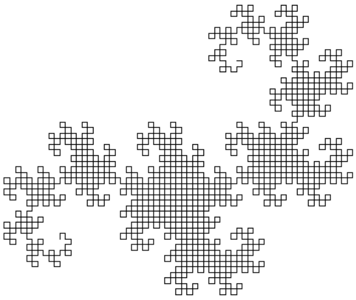
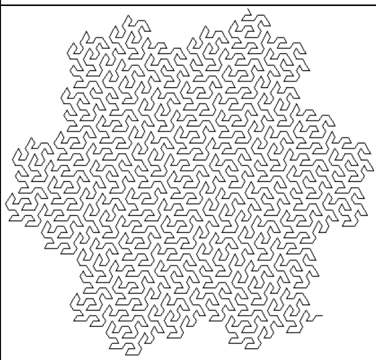
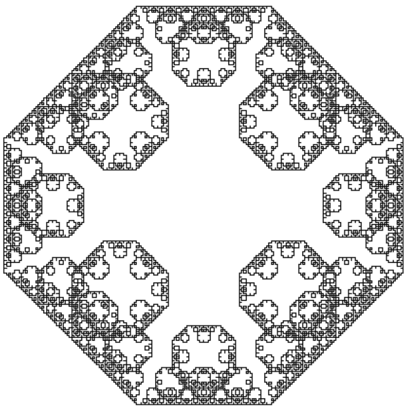
Крива Коха(трикутна)		D =ln4/ln3 = 1.2618... N=4, b=1/3
Видозмінена крива Коха (прямокутна)		Одиничний сегмент має три відрізка довжиною 1/3. два відрізка довжиною 1/4. Зі співвідношення $3(1/3)^D + 2(1/4)^D = 1$ D= 1.34.
Видозмінена крива Коха (меандр)		D =ln8/ln4 = 3/2 N=8, b=1/4
Канторівський пил		D =ln2/ln3 = 0,6309... N=2, b=1/3

Дерево		$D = \ln 5/\ln 3$ = 1,465... $N=5$, $b=1/3$
Сніжинка		$D = \ln 3/\ln(3)^{1/2}$ = 2 $N=3$, $b=1/(3)^{1/2}$
Прокладка Серпінського		$D = \ln 3/\ln 2$ = 1,585... $N=3$, $b=1/2$

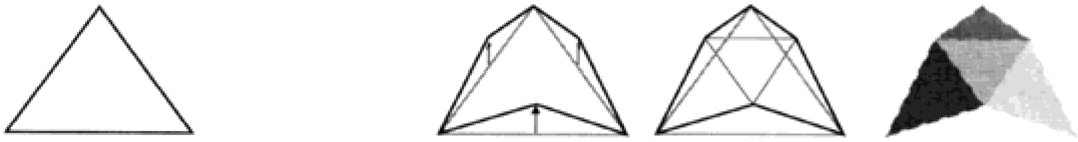
Приклади таких кривих (табл. 2):

Таблиця 2

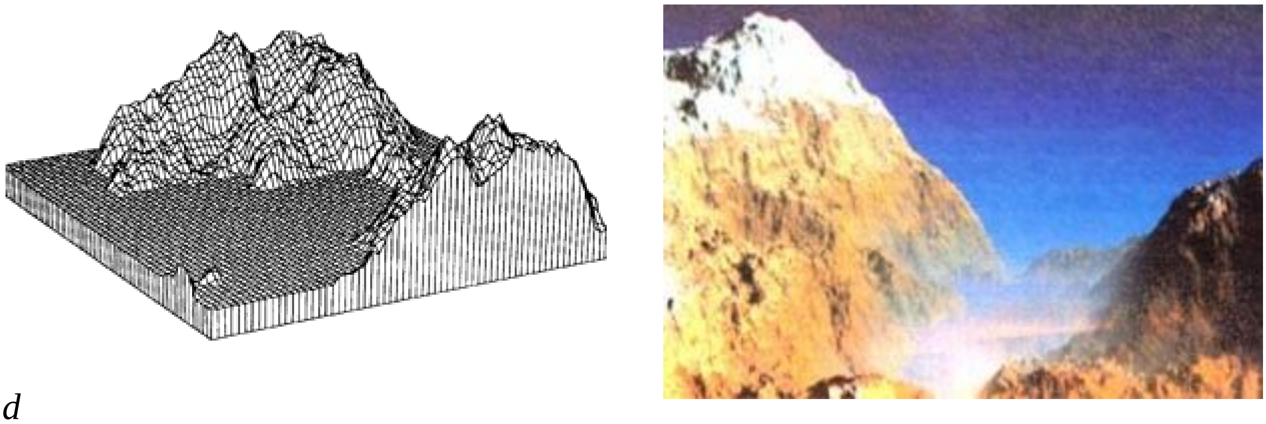
Приклади кривих

крива дракона		Аксіома: FX Правила: X → X+YF+ Y → -FX-Y $\theta = \frac{\pi}{2}$
крива Госпера		Аксіома: XF Правила: X → X+YF++YF-FX--FXFX-YF Y → -FX+YFYF++YF+FX--FX-Y $\theta = \frac{\pi}{3}$
крива Леві		Аксіома: F++F+ Правило: F → -F++F- $\theta = \frac{\pi}{4}$

Один з фракталів випадкового типу може починатися з трикутника, що лежить в довільній площині. Середні точки сторін трикутника з'єднані між собою, так що трикутник виявляється розділеним на чотири менших трикутника. Потім кожна середня точка зсувається вгору або вниз на певну, випадково обирану величину. Той самий процес застосовується до кожного з менших трикутників, потім до ще менших і так далі до нескінченності. Після чималої кількості ітерацій починає виникати все більш деталізована поверхня [3]. У цьому методі зміщення середніх точок випадкові величини для переміщення середніх точок вгору або вниз управляються певним законом розподілу, який ретельно підбирається, щоб отримати близьку апроксимацію бажаної поверхні. Для того щоб поверхня була відносно гладкою, в перетворення слід вбудувати правило, згідно з яким величина зміщення середніх точок повинна ставати дуже малою вже після кількох перших ітерацій. Таке правило дозволяє додавати лише невеликі «купини» до загальних обрисів ландшафту. Для представлення порізаної поверхні, характерної, скажімо, для гірського хребта або берегової лінії, більш відповідним буде правило повільного зменшення зсувів після кожного кроку ітераційного процесу. У даного методу побудови поверхонь існує багато додатків. Він застосовувався, зокрема, в якості моделі ерозії грунту, для аналізу сейсмічних явищ, щоб краще зрозуміти характер змін в зоні розломів. Р. Восс в Дослідницькому центрі корпорації IBM, скористався ідеєю методу, щоб будувати зображення планет, супутників, хмар і гірських хребтів, які мають дуже реалістичний вигляд (рис. 3).



a *b* *c*



d

Рис. 3. Створення фрактальних ландшафтів методом випадкового зміщення середньої точки.

Середні точки сторін трикутника (*a*) зміщуються вгору або вниз від площини зображення і з'єднуються з вершинами (*b*). При цьому виникає чотири менших трикутника, до яких повторно застосовується та ж процедура. Функція розподілу ймовірності визначає величину зсуву і, отже, ступінь гладкості фрактального ландшафту. Потім графічна програма комп'ютера зафарбовує трикутники, створюючи різні відтінки (*c*). В результаті отримуємо дуже реалістичну картину.

СПИСОК ЛІТЕРАТУРИ

1. Мандельброт Б. Фрактальна геометрія природи / Бенуа Мандельброт. – Москва : Інститут комп'ютерних досліджень. – 2002. – 656 с.

2. H.-O. Peitgen, H. Jurgens, & D. Saupe, Fractals for the Classroom, Parts 1-2, Introduction to Fractals and Chaos, Springer-Verlag, New York, 1992

3. Вурста С.Ю., Літнарович Р.М. Побудова фрактальних поверхонь в комп’ютерній графіці. МЕГУ, Рівне, 2010,-250 с.